Bi-Directional Training for Adaptive Beamforming and Power Control in Interference Networks

Changxin Shi, Randall A. Berry, *Member, IEEE,* and Michael L. Honig *Fellow, IEEE*

Abstract

We study distributed algorithms for adapting transmit beamformers and linear receiver filters in a Time-Division Duplex Multiple-Input Multiple-Output (MIMO) interference network. Each transmitter transmits a single beam, and neither the transmitters nor receivers have *a priori* Channel State Information (CSI). Given a fixed set of powers, we present an adaptive version of the Max-SINR algorithm: pilot symbols are alternately transmitted in the forward direction (transmitters to receivers) and in the reverse direction (receivers to transmitters). Unlike previous channel estimation schemes, transmissions in each direction are synchronized across the source or destination nodes, and the pilots are used to update the filters/beams directly using a least squares criterion. To improve the performance with limited training, we include exponential weighting of the least squares objective across data frames. In addition, bi-directional training can be used to implement *analog interference pricing* for power control: training in the forward direction is used to measure received Signal-to-Interference Plus Noise Ratios (SINRs) and interference prices, and those estimates combined with synchronous backward training are used to update the powers. Given sufficient training this method achieves the same performance as interference pricing updates with perfect CSI. Numerical results are presented that illustrate the performance of these methods in different settings.

R. A. Berry and M. L. Honig are with the Department of Electrical Engineering and Computer Science (EECS), Northwestern University, Evanston, IL 60208 USA (e-mail: rberry@eecs.northwestern.edu; mh@eecs.northwestern.edu). C. Shi was with the Department of EECS, Northwestern University. He is now with Barclays, New York, NY (e-mail: cshi@u.northwestern.edu).

This research was supported in part by ARO under grant W911NF-06-1-0339 and DARPA under grant W911NF-07-1-0028. This paper was presented in part at the 44th Conference on Information Sciences and Systems, Princeton, New Jersey, March 2010.

I. Introduction

Adding antennas to nodes in a wireless interference network provides the opportunity to increase spectral efficiency by avoiding and aligning interference [1], [2]. That requires joint optimization of the transmitter precoding matrices and transmit powers with the receive filters. Several algorithms have been proposed for this optimization assuming that the transmitters have perfect Channel State Information (CSI) using different objectives such as rate/Signal-to-Interference Plus Noise Ratio (SINR) and mean squared error (MSE) (see [2]–[13]). Here we relax the assumption that CSI is available *a priori*, and develop *adaptive* algorithms for Multi-Input/Multi-Output (MIMO) interference networks. The algorithms are *distributed* in the sense that each transmitter updates its own beam autonomously based on received pilot signals.

In the absence of CSI, one method for updating transmit beams in a MIMO interference network is to estimate first the CSI at the receivers, and then distribute it among the transmitters. This can be accomplished in several ways (e.g., see [14]–[17]), but in general, to optimize its beam autonomously each transmitter must know the CSI for all links in the network.[1] Exchanging this information may become difficult as the network size grows.[2]

We propose an alternative approach to joint beamformer-receiver optimization in a Time-Division Duplex (TDD) system based on the Max-SINR algorithm presented in [2]. (Earlier related work on alternating transmitter-receiver updates is presented in [4], [18]–[20].) Specifically, blocks of pilots are alternately transmitted in the forward direction (from transmitters to receivers), and in the reverse direction (from receivers to transmitters). In each direction the corresponding filters or beams are updated directly using a least squares criterion. In the reverse direction the complex conjugate of the receive filters are used as the beams, and because the system is TDD, channel reciprocity is assumed to hold. As the amount of training increases, each update then converges to the corresponding Max-SINR update.

Forward and backward training has also been proposed to estimate CSI in cellular networks, e.g., see [20]–[26]. A key difference here is that we directly estimate the optimal beamformer and receive filter as opposed to estimating the CSI as an intermediate step. Furthermore, in the channel estimation approach, pilot transmissions must be coordinated across nodes to avoid interference, which degrades the

[1]For joint transmitter-receiver optimization each transmitter must also know how each receive filter is computed.

[2]Alternatively, in a *centralized* scheme, all CSI could be relayed to a remote processor, which computes all beams and receive filters, and then sends those to the respective transmitters and receivers. (See the discussion in Section III-C.) Here our focus is on distributed schemes.

channel estimates. In contrast, in the proposed scheme *synchronous* transmission of pilots from *all* of the corresponding nodes in each direction is needed so that the interference is included in the estimated covariance matrices. Each transmitter then updates its precoder autonomously given the set of received pilot signals sent from the receiver nodes. (Similarly, the receiver filters are updated from the transmitted pilots in the forward direction.) Synchronous transmission of pilots combined with TDD and channel reciprocity therefore eliminates the need to estimate CSI explicitly, and to determine what CSI is needed at each node.

Numerical results are presented to illustrate the performance of the bi-directional training scheme as a function of training length and forward-backward iterations. Throughout we assume that each transmitter transmits a single beam, although the general approach can be easily extended to the scenario in which the transmitters may transmit multiple beams. With correlated block fading and short coherence blocks it is desirable to make use of pilots over many coherence blocks to reduce the estimation error. For that scenario we modify the least squares objective to include exponentially weighted data from previous blocks. That can provide a substantial improvement in sum rate when training is severely limited.

The previous scheme to adjust the beamformers assumes fixed powers. This yields good sum-rate performance provided that the system is not overloaded (i.e., interference limited at high SNRs). Otherwise, a power control scheme is needed to mitigate interference further, e.g., by turning users off. We present an adaptive version of the distributed interference pricing algorithm presented in [27]–[29] in which the interference prices and powers are updated via bi-directional training. Namely, forward training is used to estimate the received SINR and interference price, and backward training is used to estimate the interference cost at each transmitter, where the transmit powers are given by the interference prices. Powers are then updated according to a best response as in [27], [28]. This *analog pricing* scheme can be used as an outer loop for adjusting powers in combination with an inner loop for adjusting the beams and receivers.[3] Numerical results illustrate that this method can achieve the maximum high-SNR slope for sum rate (degrees of freedom) starting from an overloaded system.

In the next section, we present our system model. Bi-directional training for beam and receiver updates is presented in Section III along with numerical examples illustrating performance. Bi-directional training for power control based on analog pricing is presented in IV along with results illustrating the performance of joint power control with beamforming.

[3]A similar scheme for updating powers via backward transmissions is presented in [20, Section 6.6.2], although it is not part of a bi-directional training scheme.

II. System Model

We consider a peer-to-peer wireless network with K transmitter-receiver pairs communicating through MIMO links sharing the same spectrum. Each transmitter has N_T antennas, each receiver has N_R antennas, and the channel from the k-th transmitter to the j-th receiver is denoted by a complex matrix $\mathbf{H}_{jk} \in \mathbb{C}^{N_R \times N_T}$. Each channel is time-varying in general, but is assumed to be fixed for each coherence block, denoted by $\mathbf{H}_{jk}^{(n)}$ where n is the block index. (The index n will often be omitted when there is no confusion.) Neither the transmitters nor receivers have *a priori* channel information.

For simplicity, we assume each transmitter transmits only one beam to its desired receiver, i.e., the precoding matrix has rank one. The beamforming vector for transmitter k is $\mathbf{v}_k \in \mathbb{C}^{N_T}$ and satisfies the power constraint $\|\mathbf{v}_k\|^2 \leq P_k^{\max}$. The received signal vector at the k-th receiver in the n-th block is then

$$\mathbf{y}_k = \mathbf{H}_{kk}^{(n)} \mathbf{v}_k x_k + \sum_{j \neq k} \mathbf{H}_{kj}^{(n)} \mathbf{v}_j x_j + \mathbf{n}_k \tag{1}$$

where x_k is the unit variance data symbol from transmitter k and $\mathbf{n}_k$ is the additive noise with covariance matrix $E[\mathbf{n}_k \mathbf{n}_k^{\mathrm{H}}] = \mathbf{R}_k$.

We assume linear receivers so that the estimated symbol for user k is $\widehat{x}_k = \mathbf{g}_k^{\mathrm{H}} \mathbf{y}_k$, where $\mathbf{g}_k$ is the corresponding receive filter. Given a set of beamformers and receive filters $\{\mathbf{v}_k, \mathbf{g}_k\}$, the SINR for user k can be written as

$$\gamma_k = \frac{|\mathbf{g}_k^{\mathrm{H}} \mathbf{H}_{kk}^{(n)} \mathbf{v}_k|^2}{\sum_{j \neq k} |\mathbf{g}_k^{\mathrm{H}} \mathbf{H}_{kj}^{(n)} \mathbf{v}_j|^2 + |\mathbf{g}_k^{\mathrm{H}} \mathbf{R}_k \mathbf{g}_k|}. \tag{2}$$

Ideally, for each channel realization (block), we would like to choose the set $\{\mathbf{v}_k \, \mathbf{g}_k\}$ to maximize the sum rate subject to average power constraints. That is, our problem is

$$\max_{\mathbf{v}_1, \cdots, \mathbf{v}_K, \mathbf{g}_1, \cdots, \mathbf{g}_K} \sum_{k=1}^{K} \log(1 + \gamma_k) \tag{P1}$$
$$\text{s.t.} \quad \|\mathbf{v}_k\|^2 \leq P_k^{\max} \quad \forall\, k.$$

Writing $\mathbf{v}_k = \bar{\mathbf{v}}_k \sqrt{p_k}$, where $\bar{\mathbf{v}}_k$ is the normalized beamformer and $p_k = \|\mathbf{v}_k\|^2$ is the transmitted power for user k, we seek an algorithm that chooses both the beam direction $\bar{\mathbf{v}}_k$ and the power p_k to maximize the sum rate. This is complicated by the assumption that the channels are time-varying, and initially unknown at all nodes. Estimating the channels requires overhead, which reduces the rate, and furthermore, if the channels vary too quickly, then the channel estimates are likely to be inaccurate. Therefore we desire an estimation scheme with minimal overhead, and which adapts to the time-variations of the channel.

To simplify problem (P1), we separate the optimization of the directions and powers. That is, in the next section we assume fixed power $p_k = P_k^{\max}$ for each user, and optimize the beamformers and receivers. Then in Section IV, the power p_k is adapted to maximize the sum rate (or utility) when the directions of all beamformers and receive filters are fixed. This separation is motivated by prior work on joint optimization of powers and beamformers with perfect CSI [13]. There the Max-SINR algorithm is used in an inner loop to adapt beam directions with fixed powers, and interference pricing is used in an outer loop to adapt the powers with fixed beamformers. It is observed in [13] that this method generally performs better than alternating transmitter-receiver optimization according to a sum Minimum Mean Squared Error (MMSE) criterion, which jointly updates powers and beamformers.

III. Forward-Backward Adaptation

In this section we assume that the powers are fixed and jointly optimize the beamformer directions and receiver filters. We first review forward-backward updates with perfect CSI (bi-directional *optimization*) in Sec. III-A before introducing the adaptive version of those updates with training.

A. Bi-Directional Optimization: Max-SINR Algorithm

The Max-SINR algorithm iteratively optimizes the transmit precoders and receivers, assuming the transmitters/receivers know their direct- and cross-channel matrices. It iterates the following steps: (i) Fix the precoders and optimize the receivers; (ii) Reverse the direction of transmission, so that the roles of the receiver filters and precoders are swapped, and optimize the precoders (now the receivers).

The optimization criterion in each step is the associated SINR, i.e., in step (i) the receiver for user k is obtained by solving

$$\max_{\mathbf{g}_k} \quad \frac{|\mathbf{g}_k^{\mathrm{H}}\mathbf{H}_{kk}\mathbf{v}_k|^2}{\sum_{j\neq k}|\mathbf{g}_k^{\mathrm{H}}\mathbf{H}_{kj}\mathbf{v}_j|^2+|\mathbf{g}_k^{\mathrm{H}}\mathbf{R}_k\mathbf{g}_k|} \qquad \text{s.t.} \qquad \|\mathbf{g}_k\|^2 = P_k^{\max} \tag{3}$$

for fixed $\mathbf{v}_j$, $j=1,\cdots,K$, and in step (ii) the beamformer for user k is updated by solving

$$\max_{\mathbf{v}_k} \quad \frac{|\mathbf{v}_k^{\mathrm{H}}\mathbf{H}_{kk}^{\mathrm{H}}\mathbf{g}_k|^2}{\sum_{j\neq k}|\mathbf{v}_k^{\mathrm{H}}\mathbf{H}_{jk}^{\mathrm{H}}\mathbf{g}_j|^2+|\mathbf{v}_k^{\mathrm{H}}\mathbf{R}_k\mathbf{v}_k|} \qquad \text{s.t.} \qquad \|\mathbf{v}_k\|^2 = P_k^{\max} \tag{4}$$

for fixed $\mathbf{g}_j$, $j=1,\cdots,K$.[4] Although inspired by a duality type of argument, which applies to the uplink/downlink [30], [31], the Max-SINR method does not appear to maximize a particular objective. Hence so far, there is no proof that the algorithm converges. Nevertheless, numerical results show that

[4]We assume that the noise covariance matrix $\mathbf{R}_k$ is the same at the transmitter and receiver. That may not be valid if the noise includes interference from other transmitters in the network that are not training synchronously with the set of K transmitters considered. For example, those other transmitters may belong to a separate cluster of nodes. **While the algorithms in the sequel can still be applied in that scenario, substantial variations in background noise levels could affect performance.

for the scenarios considered with one beam per user, the Max-SINR algorithm essentially achieves the maximum sum rate over a wide range of SNRs [2], [13].

Closely related to the Max-SINR algorithm is alternating optimization of beamformers and receivers according to a sum MMSE objective [13]. Although the beamformer updates are nearly the same as in the Max-SINR algorithm (differing only in the diagonal loading of the covariance matrix), and it is straightforward to show that the MMSE objective converges, numerical examples in [13] show that the Max-SINR algorithm generally performs better at high SNRs under large loads. Hence in what follows we restrict our attention to the Max-SINR algorithm for beam updates.

B. Bi-Directional Training

Maximizing the received SINR in (3) and (4) is equivalent to minimizing the MSE at the output of the corresponding filter. This leads to an adaptive version in which the MSE is replaced by a Least Squares (LS) cost function. Here we assume that in each step the set of transmitters or the set of receivers synchronously transmit training sequences in the corresponding direction.

Referring to Figure 1, a sequence of TDD data frames (or blocks) is transmitted. Each frame is divided into a segment for training and a segment for data transmission. The training segment may consist of multiple forward and backward updates. (The backward update is shown first since the last forward update can be combined with the data transmission segment in decision-directed mode.) For the forward update the transmitters synchronously transmit a sequence of M training symbols given by the matrix $\mathbf{B}^{\mathrm{H}}$ where $\mathbf{B} = [\mathbf{b}_1^{\mathrm{H}}, \cdots, \mathbf{b}_K^{\mathrm{H}}]$, $\mathbf{b}_k$ is the $1 \times M$ row vector containing the training symbols $b_k(1), \cdots, b_k(M)$ for user k, and the superscript "H" denotes complex conjugate transpose. The received signal at receiver k is then given by (1) where $x_k(i) = b_k(i)$. At receiver k the estimated symbol at time i is then $\hat{b}_k(i) = \mathbf{g}_k^{\mathrm{H}} \mathbf{y}_k(i)$. The corresponding sequence of estimated symbols is $\widehat{\mathbf{b}}_k = \mathbf{g}_k^{\mathrm{H}} \mathbf{Y}_k$, where

$$\mathbf{Y}_k = \Big[\mathbf{y}_k(1), \cdots, \mathbf{y}_k(M)\Big] \tag{5}$$

$$= \mathbf{H}_{kk}^{(n)} \mathbf{v}_k \mathbf{b}_k + \sum_{j \neq k} \mathbf{H}_{kj}^{(n)} \mathbf{v}_j \mathbf{b}_j + \mathbf{N}_k. \tag{6}$$

where $\mathbf{N}_k = [\mathbf{n}_k(1), \cdots, \mathbf{n}_k(M)]$. The filter $\mathbf{g}_k$ is then selected to minimize $\|\mathbf{b}_k - \mathbf{g}_k^{\mathrm{H}} \mathbf{Y}_k\|^2$, which gives

$$\mathbf{g}_k = (\mathbf{Y}_k \mathbf{Y}_k^{\mathrm{H}})^{-1} \mathbf{Y}_k \mathbf{b}_k^{\mathrm{H}}. \tag{7}$$

We refer to this as *forward training*, and to (7) as the *forward update*.

The beamformers $\mathbf{v}_1, \cdots, \mathbf{v}_K$ are similarly updated via *backward training* exploiting channel reciprocity. Specifically, the reverse channel from receiver k to transmitter j is $\overleftarrow{\mathbf{H}}_{jk} = \mathbf{H}_{kj}^{\mathrm{T}}$, where the

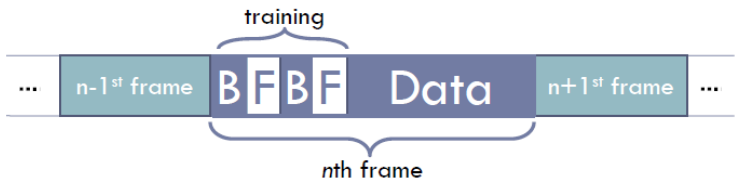

Fig. 1. Sequence of TDD frames with bi-directional training. The training period in the nth frame shows two backward-forward iterations followed by data transmission.

superscript "T" denotes transpose. Fixing the set of (original) receive filters $\{\mathbf{g}_k\}$, receiver k then applies $\mathbf{g}_k^*$ as the beamformer, where "$*$" denotes complex conjugate, and all receivers (sink nodes) synchronously transmit training sequences in the reverse direction.[5] Let $\overleftarrow{\mathbf{b}}_k$ denote the training sequence from receiver k. Then the observed signal at transmitter k is given by

$$\overleftarrow{\mathbf{Y}}_k = \mathbf{H}_{kk}^{(n)\,\mathrm{T}} \mathbf{g}_k^* \overleftarrow{\mathbf{b}}_k + \sum_{j\neq k} \mathbf{H}_{jk}^{(n)\,\mathrm{T}} \mathbf{g}_j^* \overleftarrow{\mathbf{b}}_j + \overleftarrow{\mathbf{N}}_k \tag{8}$$

where $\overleftarrow{\mathbf{N}}_k = \left[\overleftarrow{\mathbf{n}}_k(1), \cdots, \overleftarrow{\mathbf{n}}_k(M)\right]$ is the vector of M independent noise samples.

Note that $\overleftarrow{\mathbf{Y}}_k^{\,*}$ (instead of $\overleftarrow{\mathbf{Y}}_k$) corresponds to the reverse signal used to compute the SINR in the Max-SINR algorithm, where the transmitted symbol $x_k = \overleftarrow{\mathbf{b}}_k^{\,*}$. Hence replacing the corresponding MSE by the LS cost function, we wish to select $\mathbf{v}_k$ to

$$\min_{\mathbf{v}_k} \quad \|\overleftarrow{\mathbf{b}}_k^* - \mathbf{v}_k^{\mathrm{H}} \overleftarrow{\mathbf{Y}}_k^*\|^2 = \|\overleftarrow{\mathbf{b}}_k - \mathbf{v}_k^{\mathrm{T}} \overleftarrow{\mathbf{Y}}_k\|^2 \tag{9}$$

giving the solution

$$\mathbf{v}_k = \left((\overleftarrow{\mathbf{Y}}_k \overleftarrow{\mathbf{Y}}_k^{\mathrm{H}})^{-1} \overleftarrow{\mathbf{Y}}_k \overleftarrow{\mathbf{b}}_k^{\mathrm{H}} \right)^*, \tag{10}$$

which is the *backward update*. We must normalize the beamformer/receive filter after each update to satisfy the power constraint. This scaling does not change the SINR of the corresponding reverse/forward link. However, it does influence the results in subsequent updates. (This normalization is also included in the Max-SINR algorithm and has been empirically observed to improve performance relative to unnormalized updates.)

The *bi-directional* adaptive LS beamforming algorithm therefore consists of the following steps:

1) *Backward training*: The receivers synchronously transmit M training symbols given by the backward training matrix $\overleftarrow{\mathbf{B}}$. Receiver k uses the current estimate $\mathbf{g}_k^*$ as the beamformer, and each

[5] We use $\mathbf{g}_k^*$ instead of $\mathbf{g}_k$ in the reverse direction since otherwise we would have to pass the backward signal through the conjugate of the channel instead of the channel itself.

transmitter k updates the beamformer $\mathbf{v}_k$ according to (10) with a normalization to satisfy the corresponding power constraint.

2) *Forward training*: The transmitters synchronously transmit M training symbols given by the training matrix $\mathbf{B}$, and each receiver k updates the filter $\mathbf{g}_k$ according to (7) with a normalization to satisfy the reverse link power constraint.
3) Iterate the preceding steps up to a maximum number of iterations, or until a convergence criterion is satisfied.
4) Transmit data in the forward direction.

These updates are repeated every frame during the training period. If the set of users or the channels change significantly from frame to frame, then the order of the forward-backward updates does not significantly affect the performance (i.e., convergence of sum rate). However, if the users are fixed and the channels across frames vary slowly, then the receivers can continue to train during the data phase in decision-directed mode. In that scenario it is best to start each frame with backward training using the optimized (updated) receivers as beams. Similarly, initializing with a backward update may be best if the receivers have estimated CSI from previous transmissions. For example, if the receivers know the direct channel, then the beams used for backward training can be the optimized beams in the absence of interference. Otherwise, with no *a priori* CSI the transmitters may initialize by transmitting pilots through random beams.

The training sequences must be linearly independent across transmitters/receivers in order to distinguish all sources, and ideally should have low cross-correlation to improve the estimation accuracy. As the training length M becomes large, the solution given by (7) and (10) approaches the corresponding MMSE solution, or equivalently, the update in the Max-SINR algorithm. Hence by running sufficiently many forward-backward cycles within each frame, each with sufficiently long training sequences, the performance should approach that of the Max-SINR algorithm.

Given a fixed amount of training data there is generally an optimal number of forward-backward iterations. With too few iterations the transmit beamformers and receive filters are not close to the fixed point, whereas with too many iterations each segment contains insufficient training symbols to obtain accurate filter estimates. This tradeoff is illustrated in Section III-E. As the SNR increases, the trade-off generally favors more iterations since the number of iterations needed to achieve the optimal fixed point increases with SNR.

C. Comparison with Two-Way Channel Estimation

As noted previously, forward-backward training or *two-way estimation* schemes have also been proposed for channel estimation (e.g., see [21]–[26])). Here we contrast that approach with the bi-directional LS algorithm presented here. A key difference is that the LS updates (7) and (10) directly estimate the filters instead of estimating the channels that are then substituted in the MMSE expression. Direct filter estimation has the attractive property of automatically accounting for varying interference levels and filter estimation error. That is, pilots from distant transmitters/receivers have little effect on the filter estimate, so are automatically ignored.

In contrast, channel estimation schemes must determine what CSI needs to be estimated and exchanged. This is straightforward in a single-cell within a cellular system where CSI for all users in the cell is needed. However, deciding on what CSI is important becomes an issue with multi-cell cooperation. There it is necessary to stagger the transmission of pilots across cells to avoid "pilot contamination" [32], [33]. In addition to the design problem of coordinating these transmissions, this may extend the training period.

Whereas bi-directional LS training is a distributed scheme (i.e., each transmitter or receiver computes its beam or filter autonomously), the channel estimation approach is best suited for centralized updates. That is, in the channel estimation approach, to update its beam each transmitter must know the CSI across the *entire* network (or at least from all strong interferers), including direct channels for interfering transmitter-receiver pairs. In a distributed implementation all CSI would therefore have to be relayed to all other transmitter nodes, and each transmitter would have to solve essentially the same optimization problem for all beams and receivers in the network. Of course, a more efficient approach would be to relay all CSI to a single remote processor that computes all beams, which are then relayed back to the appropriate transmitters. (See, for example, the scheme proposed in [15], which could be used with this architecture.) While that approach may be appropriate for some applications, in other applications (including peer-to-peer and ad-hoc networks) a distributed algorithm for beam updates may be desirable.

Another attractive property of LS filter estimation is that it provides the best filter estimate at the transmit/receive side given the current set of filters/beams at the opposite side. In contrast, in the channel estimation approach filter estimates may be modified to account for inaccurate CSI [22]. Finally, comparing bi-directional LS filter estimation with *centralized* channel estimation, a disadvantage of bi-directional estimation is that it takes multiple iterations to converge.[6]

[6]Note, however, that the amount of training required to estimate all of the channels in the network would be far more than that needed to update a few antenna weights in each forward-backward iteration.

D. Short Coherence Blocks: Recursive Block Least Squares

For a given coherence block length L, assumed to be the size of each frame, there is an optimal amount of training per block; more training gives better filter estimates, but takes away symbols for data transmission. As the length of the coherence block decreases, the optimal training length decreases. One way to effectively increase the amount of training for small L is to include training data from previous blocks. Because the channels, beams, and filters are assumed to vary over successive blocks, it is then important to discount the data from past blocks when computing the current estimates. One way to do this is to modify the least squares cost function by including exponentially weighted data from previous blocks, namely,

$$e_k^{(n)} = \sum_{l=1}^{n} \lambda^{n-l} \left(\sum_{i=1}^{M} |b_k^{(l)}(i) - \mathbf{g}_k^{\mathrm{H}} \mathbf{y}_k^{(l)}(i)|^2 \right) \tag{11}$$

where n is the current block index, $b_k^{(l)}(i)$ and $\mathbf{y}_k^{(l)}(i)$ are, respectively, the i-th training symbol and corresponding received signal vector for user k in block l, the summation in the parentheses is the sum squared error for block l, and $\lambda \in (0, 1]$ is the exponential weighting factor. Roughly speaking, the training window spans $1/(1-\lambda)$ coherence blocks. Taking $\lambda = 1$ corresponds to infinite memory.

We also add a regularization term $\delta\lambda^n \|\mathbf{g}_k\|$, where δ is a small positive constant. This helps to stabilize the solution when n is small, since the amount of training may be insufficient to estimate the filters. Given the training sequence $\mathbf{b}_k^{(l)}$ and the received signals $\mathbf{Y}_k^{(l)}$ for block $l = 1, \cdots, n$, the forward update for the receive filter of user k is obtained by solving

$$\min_{\mathbf{g}_k} \quad \sum_{l=1}^{n} \lambda^{n-l} \|\mathbf{b}_k^{(l)} - \mathbf{g}_k^{\mathrm{H}} \mathbf{Y}_k^{(l)}\|^2 + \delta\lambda^n \|\mathbf{g}_k\|^2. \tag{12}$$

The solution to this minimization problem can be computed for each block n, assuming one forward/backward update per block. However, it is not necessary to store all of the past data to update the solution. A block recursive algorithm for updating $\mathbf{g}_k$ is shown in Table I, and consists of updating the state variables $\mathbf{P}_k^{(n)}$ ($N_R \times N_R$ matrix) and $\mathbf{K}_k^{(n)}$ ($N_R \times M$ matrix) at each block using the current training data. (The derivation is given in [34].)

Similarly, in the backward direction, we update the beamformer $\mathbf{v}_k$ using the analogous exponentially weighted LS objective, i.e.,

$$\min_{\mathbf{v}_k} \quad \sum_{l=1}^{n} \lambda^{n-l} \|\overleftarrow{\mathbf{b}}_k^{(l)} - \mathbf{v}_k^{\mathrm{T}} \overleftarrow{\mathbf{Y}}_k^{(l)}\|^2 + \delta\lambda^n \|\mathbf{v}_k\|^2. \tag{13}$$

The recursive method in Table I is again applicable where each transmitter updates a matrix $\mathbf{Q}_k^{(n)}$ as the counterpart of $\mathbf{P}_k^{(n)}$. The *Bi-Directional Recursive Least Squares (RLS)* beam adaptation algorithm

TABLE I

BLOCK RECURSIVE LS ALGORITHM

Initialization
$\mathbf{g}_k^{(0)} = 0, \qquad \mathbf{P}_k^{(0)} = \delta^{-1}\mathbf{I}_{N_R \times N_R}$
For each block n, compute
$\mathbf{K}_k^{(n)} = \lambda^{-1}\mathbf{P}_k^{(n-1)}\mathbf{Y}_k^{(n)}\Big(\mathbf{I}_{M\times M} + \lambda^{-1}\mathbf{Y}_k^{(n)\,\mathrm{H}}\mathbf{P}_k^{(n-1)}\mathbf{Y}_k^{(n)}\Big)^{-1}$ $\mathbf{g}_k^{(n)} = \mathbf{g}_k^{(n-1)} - \mathbf{K}_k^{(n)}(\mathbf{b}_k^{(n)\,\mathrm{H}} - \mathbf{Y}_k^{(n)\,\mathrm{H}}\mathbf{g}_k^{(n-1)})$ $\mathbf{P}_k^{(n)} = \lambda^{-1}\mathbf{P}_k^{(n-1)} - \lambda^{-1}\mathbf{K}_k^{(n)}\mathbf{Y}_k^{(n)\,\mathrm{H}}\mathbf{P}_k^{(n-1)}$

is then given by the same steps as in the previous LS adaptive beam adaptation algorithm, where the forward updates for $\mathbf{g}_k^{(n)}$, $\mathbf{K}_k^{(n)}$, and $\mathbf{P}_k^{(n)}$ are given in Table I. The backward update for $\mathbf{v}_k^{(n)}$ is also given in Table I, where $\mathbf{v}_k^*$ and $\mathbf{Q}_k$ are substituted for $\mathbf{g}_k$ and $\mathbf{P}_k$, respectively. The first training phase can use an arbitrary set of beamformers/receive filters, including the initialization shown in Table I. In contrast with the (unweighted) LS algorithm, here we assume only one forward-backward iteration per block. This is because the training length is assumed to be relatively short, so that multiple iterations would likely degrade performance.

E. Numerical Examples

In this section, we present some numerical examples that illustrate the performance of bi-directional training with fixed powers. In the next section we present examples that illustrate the performance of combined power control with beamforming. In all examples the channel matrices (direct and cross) are independently generated with *i.i.d.* complex Gaussian elements, and are assumed to be constant within each frame (coherence block). Additive white Gaussian noise is assumed with variance σ_0^2 so that the SNR is $1/\sigma_0^2$. All results are averaged over multiple channel realizations. The examples in this section assume a network of three users with 2×2 MIMO channels. For any set of channel realizations with probability one there are two zero-forcing solutions for the beams and receive filters, which eliminate the interference [35]. All bi-directional training examples assume an additional receive filter update during the data transmission segment.[7]

[7]Specifically, each receiver applies the current filter to estimate the transmitted binary symbols, and at the end of each block updates the receive filter again to minimize the LS objective using the estimated symbols. (The improvement due to this modification is marginal.) The sum rate is evaluated at the beginning of the data transmission period for each block.

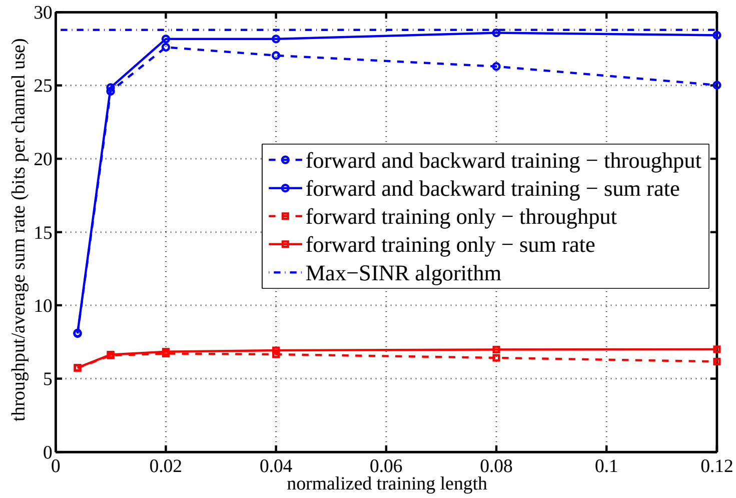

Fig. 2. Sum rate for the bi-directional LS algorithm and forward training only with constant channels.

1) Stationary Channels: Figure 2 shows the sum-rate achieved with bi-directional training as a function of the training length ($2M$) normalized by the block length L, which is 1000 symbols. In this example there is a single iteration of bi-directional training per block, and the results are shown after convergence. The results are for high SNR (30 dB), so that the optimal beams and receive filters are close to the interference aligned (zero-forcing) solution. The sum rate achieved by the Max-SINR algorithm is also shown for comparison.

As expected, as the training length becomes large, the sum-rate approaches that achieved by the Max-SINR algorithm with perfect CSI. The figure also shows the throughput achieved by the bi-directional scheme, where throughput is given by the rate per channel use after subtracting off the overhead for training. Accounting for this overhead, it can be seen that the optimal normalized training length is around 0.02 (20 symbols). For comparison, we also show the sum-rate and throughput achieved with forward training only in which the initial (randomly chosen) beamformer of each user is fixed and only the receiver filters are updated. Joint adaptation of the beams and receivers offers a dramatic improvement over forward training only.

2) Tradeoff Between Training and Iterations: To illustrate the effect of varying the number of backward-forward iterations per coherence block, we consider a network with *i.i.d.* block fading channels, i.e., the channel realizations are independently chosen each new frame. Figure 3 shows sum-rate versus total training length for an SNR of 20 dB. Each curve corresponds to a different number of iterations (cycles) per block, where the total amount of training is evenly divided among each iteration. For example, with

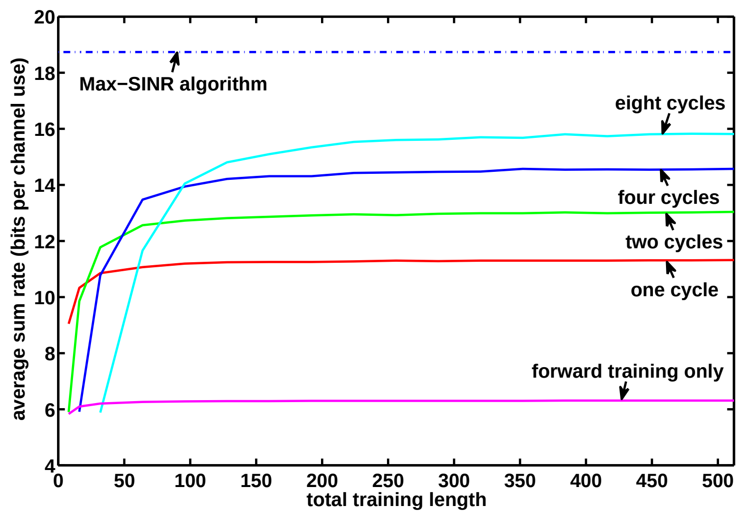

Fig. 3. Sum-rate versus training length with bi-directional training and *i.i.d.* block fading channels. Plots are shown for different numbers of forward-backward iterations (cycles).

128 training symbols and 4 bi-directional iterations, the training alternates between the forward and backward directions every 16 symbols. We again show the performance of the Max-SINR algorithm and that of forward training only. As before, bi-directional training can provide a substantial benefit relative to forward training only. However, the results also indicate that significant training is needed to approach the corresponding sum rate with perfect channel knowledge. It can also be seen that given a fixed training length there is an optimal number of bi-directional iterations; with too few iterations the transmit beam and receive filter do not converge to the appropriate fixed point, whereas with too many iterations each training segment contains insufficient training symbols to obtain accurate filter estimates. The optimal number of iterations increases with SNR.

3) Block RLS with Correlated Block Fading: The bi-directional block RLS algorithm was proposed for scenarios with short coherence blocks (small L). To illustrate the associated benefits, we consider a block fading channel model in which each channel $\mathbf{H}_{jk}$ changes from block to block according to the update

$$\mathbf{H}_{jk}^{(n)} = \alpha \mathbf{H}_{jk}^{(n-1)} + \sqrt{1-\alpha^2}\mathbf{W}_{jk}^{(n)} \tag{14}$$

where $\mathbf{W}_{jk}^{(n)}$ is a matrix with *i.i.d* complex Gaussian entries having zero mean and the same variance as $\mathbf{H}_{jk}$, and $\alpha \in [0, 1]$ is a constant that determines the correlation between successive blocks. ($\alpha = 0$ and $\alpha = 1$ correspond to *i.i.d.* block fading and stationary channels, respectively.) Given a fixed coherence

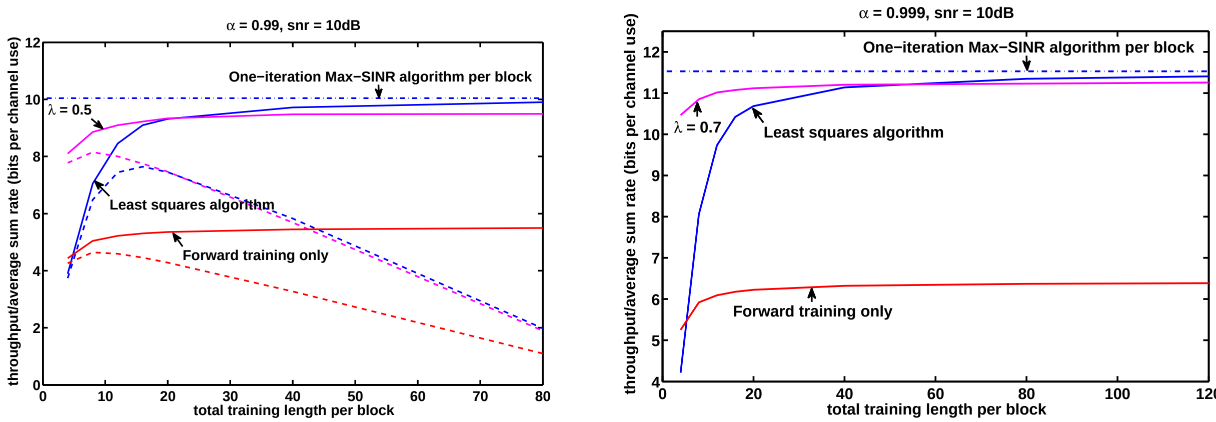

Fig. 4. Sum rate for the LS and RLS algorithms with $\lambda = 0.5$ and $\alpha = 0.99$ (left), and $\lambda = 0.7$ and $\alpha = 0.999$ (right).

block size L, the estimation error from bi-directional training is expected to increase as α decreases.[8]

Figure 4 shows the performance of the bi-directional RLS algorithm with different exponential weighting factors λ along with the unweighted LS algorithm as a function of the training length. Examples are shown for channels with different correlations corresponding to $\alpha = 0.99$ and $\alpha = 0.999$, respectively. For $\alpha = 0.99$ both the sum-rate (solid lines) and the throughput accounting for training overhead (dashed lines) are shown for each algorithm as well as for forward-training only. (For $\alpha = 0.999$ the training overhead is quite small.) For all the algorithms, a single iteration of training is used per block and the SNR is 10 dB. When the total training length is severely limited, the bi-directional RLS algorithm with an appropriate λ gives a higher sum-rate than the unweighted LS algorithm, whereas with sufficient training the LS algorithm gives the higher rate. The increase in rate from block RLS is more significant when α is closer to 1, i.e., the channel varies more slowly. When the training overhead is taken into account, the RLS gains diminish, and for $\alpha = 0.99$ become insignificant for most ranges of training. Of course, this comparison depends on the block-length L, which here is taken to be $\frac{1}{1-\alpha}$. In other simulations, we have also observed that the performance benefits of the RLS algorithm are greater at lower SNRs where estimation becomes more difficult.

[8]The degrees of freedom (high-SNR slope of achievable sum rate versus SNR) are then limited by this estimation error. This is illustrated in [34].

IV. Bi-directional Training for Power Control

For a heavily loaded system in which the sum rate is interference-limited, power control can often provide substantial improvements in the sum rate. In addition, power control can be used to implement priorities, e.g., corresponding to maximizing weighted sum rate. In this section we assume that the beamformer directions $\bar{\mathbf{v}}_k$ and receive filters $\mathbf{g}_k$ are fixed, so that the original problem (P1) reduces to a power control problem in a network with Single-Input Single-Output (SISO) links characterized by the set of *scalar* channel coefficients $\{h_{jk} = \mathbf{g}_j^{\mathrm{H}} \mathbf{H}_{jk} \bar{\mathbf{v}}_k\}$. We consider this problem where the sum rate objective is replaced by a more general sum utility objective, which can account for different weights/priorities across users. Specifically, we assume that user k is assigned the utility function $u_k(\gamma_k)$ so that Problem (P1) becomes

$$\max_{p_1,\cdots,p_K} \sum_{k=1}^{K} u_k(\gamma_k) \tag{15}$$
$$\text{s.t.} \quad p_k \leq P_k^{\max} \quad \forall\, k,$$

where similar to (2),

$$\gamma_k = \frac{p_k |h_{kk}|^2}{\sigma_0^2 + \sum_{j \neq k} p_j |h_{kj}|^2}, \tag{16}$$

assuming the normalized beamformers and receive filters are fixed. We next present an algorithm for adapting the powers based on bi-directional training, which does not assume knowledge of CSI (gains h_{kj}).

A. Interference Pricing

The power control scheme in this section is based on the *Asynchronous Distributed Pricing (ADP)* algorithm presented in [27]. In the ADP algorithm each receiver announces an *interference price*, defined as the marginal decrease in utility per marginal increase in interference, i.e.,

$$\pi_k = -\frac{\partial u_k[\gamma_k(\mathbf{p})]}{\partial I_k(\mathbf{p}_{-k})} = \frac{u_k'[\gamma_k(\mathbf{p})] p_k |h_{kk}|^2}{(\sigma_0^2 + \sum_{j \neq k} p_j |h_{kj}|^2)^2}, \tag{17}$$

where the interference power $I_k(\mathbf{p}_{-k}) = \sum_{j \neq k} p_j |h_{kj}|^2$, $\mathbf{p}$ and $\mathbf{p}_{-k}$ denote the power profiles across all users and across all users except user k, respectively, and $u_k'[\gamma_k(\mathbf{p})]$ is the derivative of the utility function. Given a set of fixed interference prices and powers for the other users, transmitter k updates its power by solving the subproblem

$$\max_{p_k} \quad u_k[\gamma_k(p_k; \mathbf{p}_{-k})] - p_k \sum_{j \neq k} \pi_j |h_{jk}|^2 \tag{P2}$$
$$\text{s.t.} \quad p_k \leq P_k^{\max}.$$

This can be viewed as user k's "best response" update for maximizing its utility minus the (linearized) interference cost to other receivers $j \neq k$.

In the ADP algorithm, the price and power updates (P2) and (17) are iterated until convergence. These updates can occur asynchronously, i.e., at arbitrary times. In the absence of CSI the interference price π_k can be estimated by obtaining estimates of the received power $p_k|h_{kk}|^2$ and the SINR γ_k. The prices and estimates for the channels h_{kj} for all j must then be relayed to the transmitters in order to update the powers. Instead, we present an alternative scheme for directly updating the powers without explicitly estimating channels and exchanging prices.

B. Analog Pricing

We first observe that if all receivers $j \neq k$ synchronously transmit independent training symbols with power π_j, then the total power measured at transmitter k is $\sum_{j\neq k} \pi_j |h_{jk}|^2$, which is the interference cost term appearing in Problem P2. Hence the interference cost can be measured by backward training where the pilot from user k is weighted by the *analog* value $\pi_k^{1/2}$.

We next show how synchronous bi-directional training can be applied to obtain the information needed for the power update, given by (P2). Specifically, the interference price can be estimated at each receiver from forward training, and both the interference cost and SINR can be estimated from backward training. We note that a similar method for estimating the interference cost in (P2) is presented in [20, Sec. 6.6.2]. However, there the estimated SINR γ_k is fed back to the transmitter through a separate control channel, and the powers are updated by a gradient algorithm as opposed to the best response updates given here.

1) Forward training (price update): From (17) we can rewrite

$$\pi_k = \frac{u_k'(\gamma_k) D_k}{(T_k - D_k)^2} \tag{18}$$

where $D_k = p_k|h_{kk}|^2$ is the desired received power and $T_k = \sum_j p_j |h_{kj}|^2 + \sigma_0^2$ is the total received power. We can estimate D_k, T_k, and the SINR γ_k from forward training. That is, the transmitters simultaneously transmit a sequence of M training symbols $\mathbf{b}_k = \{b_k(1), \cdots, b_k(M)\}$ (row vector) with power p_k, so that the signal at receiver k is

$$\mathbf{y}_k = h_{kk}\sqrt{p_k}\mathbf{b}_k + \sum_{j\neq k} h_{kj}\sqrt{p_j}\mathbf{b}_j + \mathbf{n}_k, \tag{19}$$

where $\mathbf{n}_k = \{n_k(1), \cdots, n_k(M)\}$ are M samples of white Gaussian noise.

We will use the notation $a \sim b$ to indicate that a converges to b as the training length tends to infinity, i.e.,

$$a \sim b \iff \lim_{M\to\infty} a = b.$$

In that case a can be used as an estimate for b with finite training.[9] We will assume that the $\mathbf{b}_k$'s are independent and have *i.i.d.* elements with unit magnitude.

From (19) we have

$$\mathbf{y}_k \mathbf{b}_k^{\mathrm{H}} = M h_{kk} \sqrt{p_k} + \sum_{j \neq k} h_{kj} \sqrt{p_j} (\mathbf{b}_j \mathbf{b}_k^H) + \mathbf{n}_k \mathbf{b}_k^{\mathrm{H}}, \tag{20}$$

hence

$$|\mathbf{y}_k \mathbf{b}_k^{\mathrm{H}}|^2 / M^2 \sim D_k \tag{21}$$

since the noise has zero mean and is independent of the training. Similarly, we have

$$\|\mathbf{y}_k\|^2 / M \sim T_k. \tag{22}$$

Noting that the SINR $\gamma_k = D_k/(T_k - D_k)$, we can therefore estimate the SINR as

$$\tilde{\gamma}_k = \frac{|\mathbf{y}_k \mathbf{b}_k^{\mathrm{H}}|^2}{M \|\mathbf{y}_k\|^2 - |\mathbf{y}_k \mathbf{b}_k^{\mathrm{H}}|^2}, \tag{23}$$

since $\tilde{\gamma}_k \sim \gamma_k$. The estimate for the price $\tilde{\pi}_k$ is then obtained by substituting the corresponding estimates for D_k, T_k, and γ_k in (18).

2) Backward training (power update): Given the set of estimated interference prices $\{\tilde{\pi}_k\}$, the power update requires solving Problem P2. The first-order optimality condition can be written as

$$p_k^* = \left[\frac{p_k}{\gamma_k} u_k'^{-1} \left(\left(\overleftarrow{T}_k - \overleftarrow{D}_k - \sigma_0^2 \right) \frac{p_k}{\gamma_k} \right) \right]_0^{P_k^{\max}} \tag{24}$$

where $u_k'^{-1}$ is the inverse function of u_k', $[\cdot]_0^{P_k^{\max}}$ constrains the value within the interval $[0, P_k^{\max}]$, and

$$\overleftarrow{D}_k = |h_{kk}|^2 \pi_k, \qquad \overleftarrow{T}_k = \sum_j |h_{jk}|^2 \pi_j + \sigma_0^2. \tag{25}$$

To compute the update (24) the transmitter must have estimates of $\overleftarrow{T}_k$, $\overleftarrow{D}_k$, and the SINR γ_k. In the backward training phase suppose that the receiver nodes synchronously transmit the training sequence $\overleftarrow{\mathbf{b}}_k$ (row vector) with power π_k.[10] The received signal at transmitter k (row vector) is then

$$\overleftarrow{\mathbf{y}}_k = h_{kk} \sqrt{\tilde{\pi}_k} \overleftarrow{\mathbf{b}}_k + \sum_{j \neq k} h_{jk} \sqrt{\tilde{\pi}_j} \overleftarrow{\mathbf{b}}_j + \overleftarrow{\mathbf{n}}_k, \tag{26}$$

[9] In the sequel an "estimate" for a quantity implies that this asymptotic relation is valid.

[10] In practice, π_k must be normalized to meet the peak power constraint. The performance results in Sec. IV-D ignore this constraint.

where the reverse channel from receiver k to transmitter j, $\overleftarrow{h}_{jk} = h_{kj}$ from channel reciprocity. In analogy with (21) and (22) we have

$$|\overleftarrow{\mathbf{y}}_k \mathbf{b}_k^{\mathrm{H}}|^2/M^2 \sim \overleftarrow{D}_k, \qquad \|\overleftarrow{\mathbf{y}}_k\|^2/M \sim \overleftarrow{T}_k. \tag{27}$$

To estimate the SINR from backward training we define $\eta_k(\gamma_k) = u_k'(\gamma_k)\gamma_k^2$, and note from (25), (16), and (17) that $\overleftarrow{D}_k = \eta_k(\gamma_k)/p_k$. Hence

$$\gamma_k = \eta_k^{-1}(p_k \overleftarrow{D}_k) \tag{28}$$

where p_k is the current transmitted power prior to the update. Since the function $\eta_k(\gamma_k)$ and p_k are known at the transmitter, the SINR can be estimated by replacing $\overleftarrow{D}_k$ by its estimate in (28). Finally, the noise variance σ_0^2 in (24) can be ignored if interference dominates, or can be measured when no one is transmitting, e.g., before the first power update.

The forward-backward distributed power control algorithm then consists of the following steps:

1) *Initialization:* For each user k the power $p_k = P_k^{\max}$.
2) *Forward training (price update):* The transmitters synchronously transmit M training symbols with power p_k. Each receiver estimates the received SINR $\tilde{\gamma}_k$ from (23) and computes the interference price $\tilde{\pi}_k$ from (18) where D_k and T_k are replaced by their estimates in (21) and (22).
3) *Backward training (power update):* The receivers synchronously transmit M training symbols with power $\tilde{\pi}_k$, and the transmitters then compute the estimated SINR from (28) and update their powers according to (24) where $\overleftarrow{D}_k$ and $\overleftarrow{T}_k$ are replaced by their estimates in (25).
4) Iterate steps 2 and 3.

With sufficient training the performance of this power control algorithm approaches that with perfect channel knowledge. In that scenario conditions on the utility function that guarantee convergence to either a global or local optimum are given in [27] and [29].

3) Updates with $p_k = 0$: The previous training scheme implicitly assumes that $p_k > 0$ for all k. Otherwise, if $p_k = 0$, then the SINR γ_k cannot be estimated, which prevents the price π_k from being estimated and used to determine the updated power p_k^* via backward training. Hence once a user is turned off by the preceding power control algorithm, that user cannot be turned back on again. It can happen, however, that the ADP algorithm with perfect CSI powers down a user (possibly turning that user off) in one iteration, but then increases the power in a subsequent iteration.

To accommodate this scenario, transmitter k must be allowed to transmit positive power even when the algorithm sets $p_k = 0$. However, the training signal used to estimate the channel gain h_{kk} should not be included as interference in other SINR estimates. For this purpose we introduce a second training sequence

$\mathbf{B}' = \{\mathbf{b}_1'^{\mathrm{H}}, \cdots, \mathbf{b}_K'^{\mathrm{H}}\}$, which is orthogonal to the original forward training sequence $\mathbf{B} = \{\mathbf{b}_1^{\mathrm{H}}, \cdots, \mathbf{b}_K^{\mathrm{H}}\}$, that is, $\mathbf{B}^{\mathrm{H}}\mathbf{B}' = \mathbf{0}$. During the forward training phase, if $p_k = 0$, then transmitter k transmits $\mathbf{b}_k'$ instead of $\mathbf{b}_k$ with unit power.

Each receiver $j \neq k$ then projects the observed signal $\mathbf{y}_j$ onto the space spanned by the columns of $\mathbf{B}$, giving $\tilde{\mathbf{y}}_j = \mathbf{B}\mathbf{B}^{\mathrm{H}}\mathbf{y}_j^{\mathrm{T}}/M$. This removes the contribution from transmitter k (namely, $h_{jk}\mathbf{b}_k'$) while retaining the terms from all transmitters with positive power (which transmit the corresponding training sequences in $\mathbf{B}$). Then the price can be calculated as in (18) with $\tilde{\mathbf{y}}_j^{\mathrm{T}}$ replacing $\mathbf{y}_k$ in the estimates for D_k and T_k. Since the noise power in $\tilde{\mathbf{y}}_k$ is $\|\mathbf{B}\mathbf{B}^{\mathrm{H}}\mathbf{n}_k^{\mathrm{T}}\|^2/M^2$, which has expected value $\frac{K}{M}\sigma_0^2$, the total received power can be estimated as

$$T_k = \begin{cases} \|\tilde{\mathbf{y}}_j\|^2/M + \frac{M-K}{M}\sigma_0^2 & \text{if } |\mathbf{y}_k\mathbf{b}_k'^{\mathrm{H}}|^2 < |\mathbf{y}_k\mathbf{b}_k^{\mathrm{H}}|^2 \\ \|\tilde{\mathbf{y}}_j\|^2/M + \frac{M-K}{M}\sigma_0^2 + |\mathbf{y}_k\mathbf{b}_k'^{\mathrm{H}}|^2/M^2 & \text{if } |\mathbf{y}_k\mathbf{b}_k'^{\mathrm{H}}|^2 > |\mathbf{y}_k\mathbf{b}_k^{\mathrm{H}}|^2 \end{cases}, \tag{29}$$

where the condition $|\mathbf{y}_k\mathbf{b}_k'^{\mathrm{H}}|^2 \lesseqgtr |\mathbf{y}_k\mathbf{b}_k^{\mathrm{H}}|^2$ is used to decide if transmitter k has positive power. The term containing the noise level σ_0^2 can be ignored at high SNRs. If $|\mathbf{y}_k\mathbf{b}_k'^{\mathrm{H}}|^2 > |\mathbf{y}_k\mathbf{b}_k^{\mathrm{H}}|^2$, corresponding to zero power, then the estimated ("virtual") price π_k is given by (18), where T_k is estimated by (29) and D_k is estimated as $|\mathbf{y}_k\mathbf{b}_k'^{\mathrm{H}}|^2/M^2$.

In the backward training phase, if a particular receiver k decides that its associated transmitter has $p_k = 0$, then it selects the training sequence $\mathbf{b}_k'$ instead of $\mathbf{b}_k$, as in the forward training phase, with the power equal to the virtual price. The total backward power $\overleftarrow{T}_k$ can be measured in the same way as in (29), and $\overleftarrow{D}_k$ is estimated as $|\mathbf{y}_k\mathbf{b}_k^{\mathrm{H}}|^2$ if $p_k > 0$, and as $|\mathbf{y}_k\mathbf{b}_k'^{\mathrm{H}}|^2$ if $p_k = 0$. The SINR can then be estimated using these values in (28). For users assigned zero power, we assume the SINR is estimated assuming unit transmit power corresponding to the forward training. Finally, the power can be updated using (24).

Because the projected received signal vector $\tilde{\mathbf{y}}_k$ depends on $\mathbf{B}$, each node must know the training sequences for *all* users. Hence the training matrix must be generated and distributed to all users in advance. For a system with K users at least $2K$ training symbols are needed in each direction to ensure that $\mathbf{B}$ and $\mathbf{B}'$ are orthogonal. To avoid this additional complication, we can simply exclude those users whose power is set to zero. Those users will be off permanently. Each user then only needs to know its own training sequence. However, with limited training an inaccurate estimate might lead to an undesired power update ("overreaction"), which shuts off a user prematurely. While this has been observed in numerical examples, the performance difference between these two variations is often insignificant (see Section IV-D).

C. Joint Power Control and Beamforming

Sections III and IV have presented algorithms for separately adjusting beams and powers, respectively. Here we combine these to update the beams and powers jointly. When CSI is available at each node, the Max-SINR algorithm for beamforming can be combined with interference pricing for power control [13]. Specifically, for a fixed set of transmit powers, the Max-SINR algorithm is used to update the beams in an inner loop, and the transmit powers are adjusted in an outer loop with a fixed set of beamformers and receive filters. When the current power is zero, as described in Section IV-B3, we can update the beamformer and receive filter with a unit power constraint, so that the normalized beamformer and receive filter are maintained. Alternatively, we can set that user's power to zero. Results in [13] show that the preceding algorithm turns off users when the sum rate is interference limited (i.e., zero-forcing conditions cannot be satisfied due to too many users).

When CSI is not available, we combine the bi-directional LS beam adaptation algorithm in Section III-B with the analog pricing algorithm in Section IV-B to obtain the following bi-directional joint beam and power adaptation algorithm.[11]

1) *Power update*: Given a fixed set of normalized beams $\{\bar{\mathbf{v}}_k\}$, receive filters $\{\mathbf{g}_k\}$, and transmit powers, each transmitter updates its power using the distributed analog pricing algorithm in IV-B.
2) *Beam update*: Given a fixed set of transmit powers, the beamformers and receive filters are updated using the bi-directional LS algorithm in Section III-B (i.e., according to (10) and (7)), where the beams are normalized according to the current power constraints. During the forward training phase, the training symbols are sent with the updated power, whereas in the backward training phase, the training symbols are sent with the maximum power if the current power is positive. (This is only because they do not know the current power.[12]) If the current power is zero, then that transmitter-receiver pair is inactive during this step.

These updates can be performed in any order, and the number of iterations in each step can be varied. It is then possible to perform price and power updates simultaneously with the beam/receiver updates. However, numerical examples have shown that convergence then becomes problematic. For the subsequent simulation results, the power was updated after every five forward-backward iterations for the

[11]For a MISO network, the interference pricing algorithm discussed in Section IV-A can also be used to jointly optimize the beams and powers (see [28], [29], [36]). An adaptive version is presented in [34].

[12]The numerical example presented in Section IV-D shows that the improvement in performance that can be obtained by adjusting the powers used for the backward updates is small but noticeable.

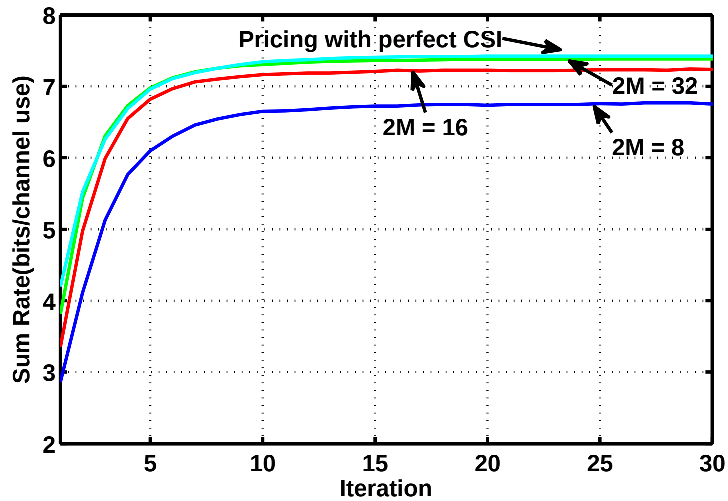

Fig. 5. Sum rate achieved by the distributed analog pricing algorithm versus iterations with different training lengths in a SISO network.

spatial filters. At high loads and high SNRs, more iterations are generally needed to update the beams and receive filters so that they are spatially aligned to avoid interference. Otherwise, the power update will turn off too many users.

D. Numerical Examples

We now present a series of numerical results that illustrates the performance of the preceding distributed analog pricing algorithms. For these examples the channels are stationary with *i.i.d.* elements, the number of users $K = 5$, and the SNR is 20 dB unless specified otherwise. All plots are averaged over 1000 random channel realizations.

1) SISO Network: Figure 5 shows the sum rate performance of the analog pricing scheme for a SISO network. In this example, it is generally optimal to turn off a subset of users. For these results when a power is set to zero, that user is permanently off. As the training length $(2M)$ increases, the performance asymptotically approaches that given by the interference pricing algorithm with perfect CSI and synchronous power and price updates, which we will refer to as the "synchronous interference pricing algorithm". For this example the training sequences were randomly generated. To distinguish the users, we must have $M \geq K$. Ideally, the training sequences in $\mathbf{B}$ should be orthogonal, but then it becomes more difficult to generate those in a distributed manner.

The performance of the analog pricing algorithm with orthogonal training sequences for zero-power

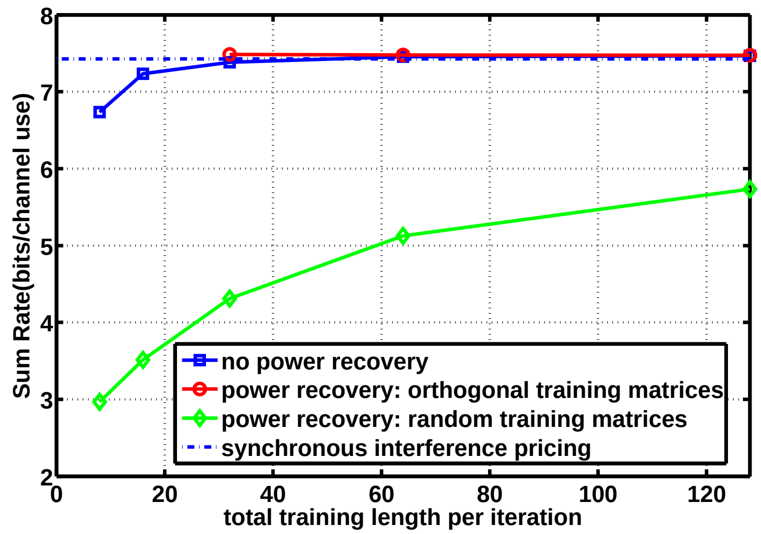

Fig. 6. Performance of the distributed analog pricing algorithm versus training length with and without orthogonal training to reactivate users.

users is shown in Figure 6. (We must have $M \geq 2K$.) Each plot shows the limiting sum rate after the algorithm converges versus training length per iteration[13]. "No power recovery" means once $p_k = 0$ it is no longer adapted, "orthogonal training matrices" means the training matrices $\mathbf{B}$ and $\mathbf{B}'$ are pre-determined and orthogonal, and "random training matrices" indicates the symbols in the training matrices are *i.i.d.* binary. In addition, we also include the performance of the synchronous interference pricing algorithm as a benchmark. Figure 6 suggests that simply excluding users with zero power in subsequent training is a reasonable strategy since the gain from introducing an additional training matrix is insignificant. The large gap between the plots for "no power recovery" and "power recovery: random training matrices" with short training lengths is due to the fact that the randomly generated training matrices $\mathbf{B}$ and $\mathbf{B}'$ are far from orthogonal, which makes it difficult to remove the interference from inactive users. Hence these results indicate that it is important to use orthogonal matrices to re-activate users.

2) MIMO Network with Joint Power and Beam Adaptation: Figure 7 shows sum rate versus iterations for the joint beam and power adaptation algorithms in Section IV-C. Each node has two antennas (2×2 channels). For these results there is no power recovery when the power is set to zero. This does not noticeably degrade performance compared with the power recovery algorithm using an orthogonal training matrix provided that there is sufficient training to generate the orthogonal training matrices. (For this

[13]The maximum number of iterations/block is 50.

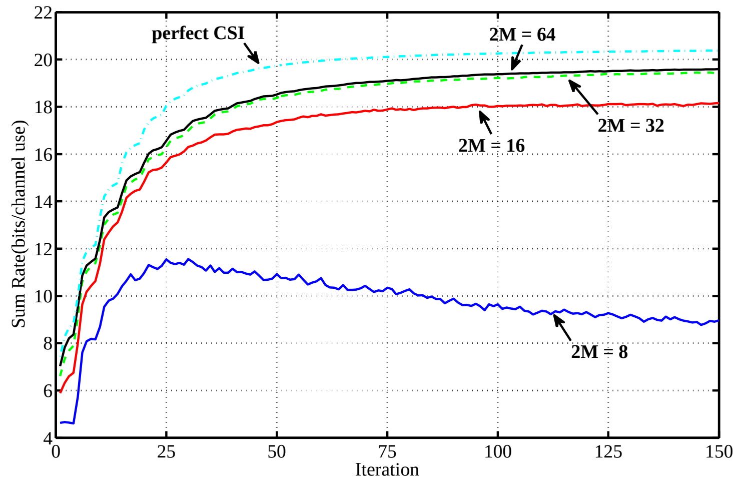

Fig. 7. Performance of the bi-directional joint beam and power adaptation algorithms for a MIMO network with 2×2 channels.

example, that requires $M \geq 10$.) The power updates are performed for all users every five iterations, which introduces the "ripple" shown in the curves (i.e., there is a more rapid increase in sum rate after each power update). Because all results are averaged over the same number of runs, the curve with $M = 4$ training samples appears noisier than the others.

Fig. 7 shows that as the training length increases, the sum rate increases, but does not asymptotically approach the sum rate achieved with perfect CSI. This is because in the backward training phase, the receiver sends the training sequence with either full or zero power, whereas when CSI is available, beams are updated according to the current power profile. When the training length for each direction is $M = 4$, the downward trend is caused by inaccurate estimation, which may prematurely turn off users. This is more likely to happen when the training length is short.

Finally, Fig. 8 shows sum rate versus SNR with joint power and beam adaptation. The plots show sum rate averaged over ten iterations after the algorithm converges. The slopes of the curves with $2M = 16$ and $2M = 32$ are nearly optimal ($3 \log_2 10$ for this example). Taking $2M = 16$ training symbols (eight in each direction) gives a loss of about 3 dB relative to perfect channel knowledge.

V. CONCLUSIONS

We have presented distributed algorithms for adapting beamformers and powers in a TDD MIMO interference network without CSI. The approach is based on bi-directional training, which can be used to update both powers and beamformers iteratively. Using an LS objective, each update approaches the

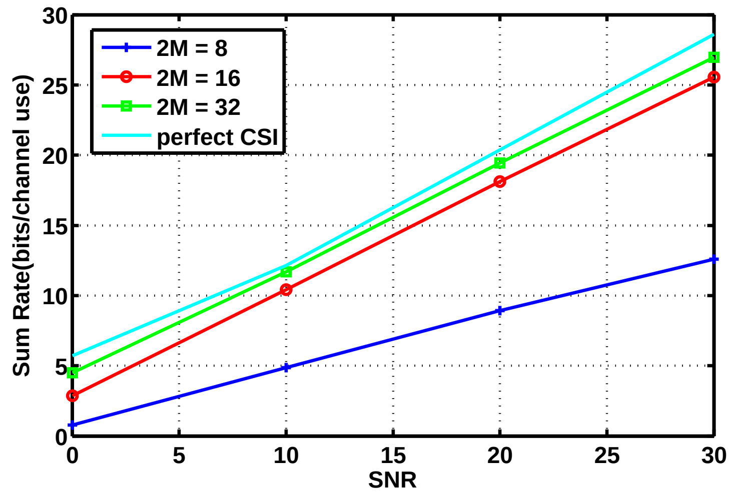

Fig. 8. Sum rate versus SNR for the bi-directional joint beam and power adaptation algorithm after convergence.

performance of the corresponding Max-SINR update with full CSI with a modest amount of training. When the channels are slowly varying across data frames, the amount of training can be reduced by including the exponentially weighted training from previous data frames. With *i.i.d.* block fading the optimal tradeoff between training length and number of forward-backward iterations depends on the bi-directional training period and the SNR. Namely, a longer training period and higher SNR shifts the tradeoff to favor more iterations.

We have also shown how bi-directional training can be used to implement adaptive power control based on the exchange of interference prices. The interference prices are estimated via forward training, and backward training is used to estimate the interference cost and feed back the SINR needed for the best response power update. The performance of the corresponding interference pricing algorithm with perfect CSI can again be achieved with a modest amount of training. This method can be easily combined with adaptive beamforming via bi-directional training; however, we have observed that joint convergence and performance is enhanced at high loads by updating the powers at a much slower rate than the beams. With sufficient training the algorithm is then observed to turn off the minimum number of users, and achieves the maximum slope for sum rate versus SNR.

The performance of all algorithms has been demonstrated assuming a peer-to-peer (interference) network with *i.i.d.* channels. Of course, bi-directional training for joint beamformer and receiver adaptation has wider applicability. For example, it can be applied to cellular networks with different antenna configurations (e.g., see [37], [38]). While an analytical characterization of performance versus training and iterations for these different scenarios appears to be difficult, it may be possible to provide some insight, e.g., using large system analysis [39]. Finally, the performance with different channel models, traffic assumptions (e.g., changing user sets due to scheduling), and possibly multiple beams per user is left for future work.

REFERENCES

[1] V. R. Cadambe and S. A. Jafar, "Interference alignment and the degrees of freedom of wireless x networks," *IEEE Trans. Inf. Theory*, vol. 55, no. 9, pp. 3893–3908, Sept. 2009.

[2] K. Gomadam, V. R. Cadambe, and S. A. Jafar, "A distributed numerical approach to interference alignment and applications to wireless interference networks," *IEEE Trans. Inf. Theory*, vol. 57, no. 6, pp. 3309–3322, June 2011.

[3] S. Ye and R. S. Blum, "Optimized signaling for MIMO interference systems with feedback," *IEEE Trans. Signal Process.*, vol. 51, no. 11, pp. 2839–2848, Nov. 2003.

[4] S. Serbetli and A. Yener, "Transceiver optimization for multiuser MIMO systems," *IEEE Trans. Signal Process.*, vol. 52, no. 1, pp. 214–226, Jan. 2004.

[5] G. Scutari, D. Palomar, and S. Barbarossa, "Competitive design of multiuser MIMO systems based on game theory: A unified view," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 7, pp. 1089–1103, Sep. 2008.

[6] S. Stanczak, M. Kaliszan, and N. Bambos, "Decentralized admission control for power-controlled wireless links," *Arxiv preprint*, vol. abs/0907.2896, 2009.

[7] C. Shi, D. A. Schmidt, R. A. Berry, M. L. Honig, and W. Utschick, "Distributed interference pricing for the MIMO interference channel," in *Proc. IEEE International Conference on Communications 2009*, Jun. 2009.

[8] S. W. Peters and R. W. Heath, "Interference alignment via alternating minimization," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Apr. 2009, pp. 2445–2448.

[9] ——, "Cooperative algorithms for MIMO interference channels," *IEEE Trans. Veh. Technol.*, vol. 60, no. 1, pp. 206–218, Jan. 2011.

[10] D. A. Schmidt, C. Shi, R. A. Berry, M. L. Honig, and W. Utschick, "Minimum mean squared error interference alignment," in *Proc. Forty-Third Asilomar Conference on Signals, Systems and Computers*, Nov. 2009, pp. 1106–1110.

[11] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Trans. Signal Process.*, vol. 59, no. 9, Sep. 2010.

[12] I. Santamaria, O. Gonzales, R. W. Heath Jr., and S. W. Peters, "Maximum sum-rate interference alignment algorithms for MIMO channels," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Miami, FL, Dec. 2010.

[13] D. Schmidt, C. Shi, R. A. Berry, M. L. Honig, and W. Utschick, "Comparison of distributed beamforming algorithms for MIMO interference networks," *IEEE Trans. Signal Process.*, vol. 61, pp. 3476–3489, Jul. 2012.

[14] M. Biguesh and A. B. Gershman, "Training-based MIMO channel estimation: A study of estimator tradeoffs and optimal training signals," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 884–893, March 2006.

[15] O. E. Ayach and J. R. W. Heath, "Interference alignment with analog channel state feedback," *IEEE Trans. Wireless Commun.*, vol. 11, no. 2, pp. 626–636, Feb. 2012.

[16] O. E. Ayach, A. Lozano, and J. R. W. Heath, "On the overhead of interference alignment: Training, feedback, and cooperation," *arXiv:1204.6100[cs.IT]*, Apr. 2012.

[17] P. Komulainen, A. Tölli, and M. Juntti, "Effective CSI signaling and decentralized beam coordination in TDD multi-cell MIMO systems," Feb. 2012, submitted to *IEEE Trans. on SP*.

[18] W. M. Jang, B. R. Vojčić, and R. Pickholtz, "Joint transmitter-receiver optimization in synchronous multiuser communications over multipath channels," *IEEE Trans. on Commun.*, vol. 46, no. 2, pp. 269–278, Feb. 1998.

[19] G. Rajappan and M. L. Honig, "Signature sequence adaptation for DS-CDMA with multipath," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 2, pp. 384–395, Feb. 2002.

[20] S. Stanczak, M. Wiczanowski, and H. Boche, *Fundamentals of Resource Allocation in Wireless Networks: Theory and Algorithms*, 2nd ed. Springer, 2008.

[21] T. L. Marzetta, "How much training is required for multiuser MIMO?" in *Proc. Fortieth Asilomar Conference on Signals, Systems and Computers*, Oct. 2006, pp. 359–363.

[22] K. S. Gomadam, H. C. Papadopoulos, and C.-E. W. Sundberg, "Techniques for multi-user MIMO with two-way training," in *Proc. IEEE International Conference on Communications*, May 2008, pp. 3360–3366.

[23] R. Osawa, H. Murata, K. Yamamoto, and S. Yoshida, "Performance of two-way channel estimation technique for multi-user distributed antenna systems with spatial precoding," in *Proc. IEEE 70th Vehicular Technology Conference*, Sep. 2009, pp. 1–5.

[24] C. Steger and A. Sabharwal, "Single-input two-way SIMO channel: Diversity-multiplexing tradeoff with two-way training," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4877–4885, Dec. 2008.

[25] L. P. Withers, R. M. Taylor, and D. M. Warme, "Echo-MIMO: A two-way channel training method for matched cooperative beamforming," *IEEE Trans. Signal Process.*, vol. 56, no. 9, pp. 4419–4432, Sep. 2008.

[26] X. Zhou, T. Lamahewa, P. Sadeghi, and S. Durrani, "Two-way training: Optimal power allocation for pilot and data transmission," *IEEE Trans. Wireless Commun.*, vol. 9, no. 2, pp. 564–569, Feb. 2010.

[27] J. Huang, R. A. Berry, and M. L. Honig, "Distributed interference compensation for wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 5, pp. 1074–1084, May 2006.

[28] C. Shi, R. A. Berry, and M. L. Honig, "Distributed interference pricing with MISO channels," in *Proc. 46th Annual Allerton Conference*, Sep. 2008.

[29] ——, "Monotonic convergence of distributed interference pricing in wireless networks," in *Proc. IEEE International Symposium on Information Theory*, Jun. 2009.

[30] N. Jindal, S. Vishwanath, and A. Goldsmith, "On the duality of Gaussian multiple-access and broadcast channels," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 768–783, May 2004.

[31] S. Vishwanath, N. Jindal, and A. Goldsmith, "Duality, achievable rates, and sum-rate capacity of Gaussian MIMO broadcast channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2658–2668, Oct. 2003.

[32] J. Jose, A. Ashikhmin, T. Marzetta, and S. Vishwanath, "Pilot contamination and precoding in multi-cell tdd systems," *IEEE Trans. on Wireless Communications*, vol. 10, no. 8, pp. 2640–2651, 2011.

[33] H. Brunner, M. Castaneda, and J. Nossek, "How much training is needed for interference coordination in cellular networks?" in *2011 International ITG Workshop on Smart Antennas (WSA)*. IEEE, 2011, pp. 1–6.

[34] C. Shi, "Distributed power and beam optimization for wireless interference networks," Ph.D. dissertation, Northwestern University, Dec. 2011.

[35] C. M. Yetis, T. Gou, S. A. Jafar, and A. H. Kayran, "On feasibility of interference alignment in MIMO interference networks," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4771–4782, Sept. 2010.

[36] D. A. Schmidt, A. Gründinger, W. Utschick, and M. L. Honig, "Distributed precoder optimization for interfering MISO Channels," in *Proc. IEEE/ITG Workshop on Smart Antennas 2008*, Feb. 2008.

[37] B. Zhuang, R. Berry, and M. L. Honig, "Interference alignment in MIMO cellular networks," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2011, pp. 3356–3359.

[38] M. Xu, D. Guo, and M. L. Honig, "Uplink-downlink bidirectional training of multiuser MIMO filters and precoders," in *Proc. Allerton Conf. on Communication, Control, and Computing*, 2011, pp. 871–878.

[39] M. J. M. Peacock, I. B. Collings, and M. L. Honig, "Performance with random signatures," in *Advances in Multiuser Detection*, M. L. Honig, Ed. Wiley Online Library, 2009, pp. 189–250.